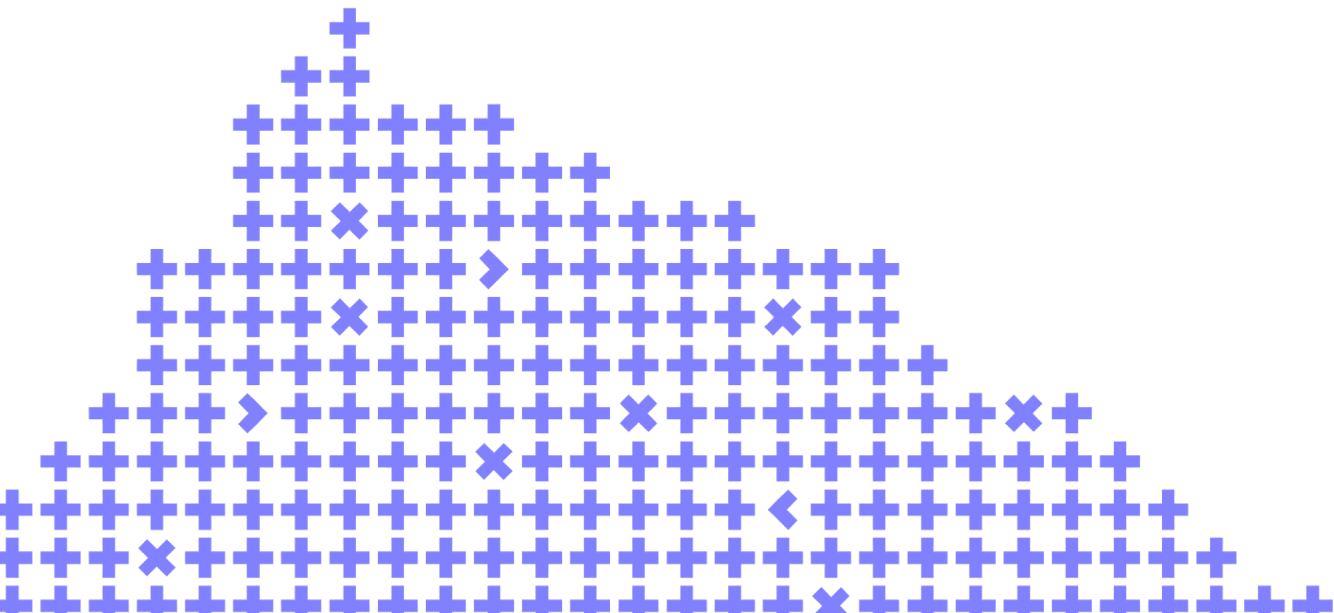


ITM as the primary KPI: pain and humiliation

Daniel Podolsky, Microavia



MICROAVIA



Co-organizer

Yandex

Who am I and why I'm qualified enough to give this speech

- 30 years of experience
 - Mostly in the go-go-go projects
 - Left last one not-allowed-to-mention-by-name in June
 - Was really disappointed almost all the time
- Some experience in the no-rush projects
 - Started in one like this in November
 - And happy

What this speech is about

- What is TTM and why is it such a buzz word

What this speech is about

- What is TTM and why is it such a buzz word
 - Short Time To Market is a way to make your business the competition winner

What this speech is about

- What is TTM and why is it such a buzz word
 - Short Time To Market is a way to make your business the competition winner
 - It is about delivering your product faster than competitors to dominate the market

What this speech is about

- What is TTM and why is it such a buzz word
 - Short Time To Market is a way to make your business the competition winner
 - It is about delivering your product faster than competitors to dominate the market
 - Frequently considered as an essential business requirement

What this speech is about

- Frequently considered as an essential requirement

What this speech is about

- Frequently considered as an essential requirement
 - Wrong!

What this speech is about

- Frequently considered as an essential requirement
 - Wrong!
- For example, MicroAvia

What this speech is about

- Frequently considered as an essential requirement
 - Wrong!
- For example, MicroAvia
 - We have a 10kg piece of Hi Tech flying over our heads and kept in the air by the inner power (battery)



What this speech is about

- Frequently considered as an essential requirement
 - Wrong!
- For example, MicroAvia
 - We have a 10kg piece of Hi Tech flying over our heads and kept in the air by the inner power (battery)
 - Of course we have Safety over the TTM



What this speech is about

- Frequently considered as an essential requirement
 - Wrong!
- For example, MicroAvia
 - We have a 10kg piece of Hi Tech flying over our heads and kept in the air by the inner power (battery)
 - Of course we have Safety over the TTM
 - This is why I'm happy to work here



What this speech is about

- Frequently considered as an essential requirement
 - Wrong!
- For example, MicroAvia
 - We have a 10kg piece of Hi Tech flying over our heads and kept in the air by the inner power (battery)
 - Of course we have Safety over the TTM
 - This is why I'm happy to work here
- So this speech is based on my previous experience

What this speech is about

- Frequently considered as an essential requirement
 - Wrong!
- For example, MicroAvia
 - We have a 10kg piece of Hi Tech flying over our heads and kept in the air by the inner power (battery)
 - Of course we have Safety over the TTM
 - This is why I'm happy to work here
- So this speech is based on my previous experience
 - Which I'm happy to leave behind

What this speech is about

- Some of the other projects are ok for TTM KPI, but simply not requiring this rush

What this speech is about

- Some of the other projects are ok for TTM KPI, but simply not requiring this rush
 - It is still common for business to insist for TTM

What this speech is about

- Some of the other projects are ok for TTM KPI, but simply not requiring this rush
 - It is still common for business to insist for TTM
 - Mostly because of desire to save money

What this speech is about

- Some of the other projects are ok for TTM KPI, but simply not requiring this rush
 - It is still common for business to insist for TTM
 - Mostly because of desire to save money
 - Or desire to speedup the business idea check cycle

What this speech is about

- Some of the other projects are ok for TTM KPI, but simply not requiring this rush
 - It is still common for business to insist for TTM
 - Mostly because of desire to save money
 - Or desire to speedup the business idea check cycle
 - Or by mistake

What this speech is about

- Finally there are some projects where short TTM could be a key to success

What this speech is about

- Finally there are some projects where short TTM could be a key to success
 - We will speak about these ones

What this speech is about

- Finally there are some projects where short TTM could be a key to success
 - We will speak about these ones
 - This speech is from CTO and team lead mostly and mostly for team leads and CTOs in startups

What this speech is about

- Finally there are some projects where short TTM could be a key to success
 - We will speak about these ones
 - This speech is from CTO and team lead mostly and mostly for team leads and CTOs in startups
- Disclaimer

What this speech is about

- Finally there are some projects where short TTM could be a key to success
 - We will speak about these ones
 - This speech is from CTO and team lead mostly and mostly for team leads and CTOs in startups
- Disclaimer
 - As it happens quite often with the speeches like this one the problem description will take more of the time than the solution description

The problem

- The problem is simple: you can not get anything for nothing, you have to sacrifice something.

The problem

- The problem is simple: you can not get anything for nothing, you have to sacrifice something.
 - Some sacrifices will be required on business side

The problem

- The problem is simple: you can not get anything for nothing, you have to sacrifice something.
 - Some sacrifices will be required on business side
 - Some on technical side

The problem

- The problem is simple: you can not get anything for nothing, you have to sacrifice something.
 - Some sacrifices will be required on business side
 - Some on technical side
 - Some on the staff side, like no war without death

The problem

- The problem is simple: you can not get anything for nothing, you have to sacrifice something.
 - Some sacrifices will be required on business side
 - Some on technical side
 - Some on the staff side, like no war without death
- I'll speak about the tech side of the story

The problem

- The problem is simple: you can not get anything for nothing, you have to sacrifice something.
 - Some sacrifices will be required on business side
 - Some on technical side
 - Some on the staff side, like no war without death
- I'll speak about the tech side of the story
 - About the damage

The problem

- The problem is simple: you can not get anything for nothing, you have to sacrifice something.
 - Some sacrifices will be required on business side
 - Some on technical side
 - Some on the staff side, like no war without death
- I'll speak about the tech side of the story
 - About the damage
 - About the ways to minimise it

The victims: Planing

- Why the planing is a victim

The victims: Planing

- Why the planing is a victim
 - Time devourer

The victims: Planing

- Why the planing is a victim
 - Time devourer
 - Businessmen's time devourer!

The victims: Planing

- Why the planing is a victim
 - Time devourer
 - Businessmen's time devourer!
 - No obvious profit

The victims: Planing

- Direct damage

The victims: Planing

- Direct damage
 - Understanding of the whole project

The victims: Planing

- Direct damage
 - Understanding of the whole project
- Collateral damage

The victims: Planing

- Direct damage
 - Understanding of the whole project
- Collateral damage
 - Agile development process

The victims: Planing

- Direct damage
 - Understanding of the whole project
- Collateral damage
 - Agile development process
 - Bus factor

The victims: Planing

- Direct damage
 - Understanding of the whole project
- Collateral damage
 - Agile development process
 - Bus factor
 - Every task is a research task

The victims: Planing

- Direct damage
 - Understanding of the whole project
- Collateral damage
 - Agile development process
 - Bus factor
 - Every task is a research task
 - Research is longer and no result guaranteed

The victims: Code quality

- What is code quality and why is it important

The victims: Code quality

- What is code quality and why is it important
 - Code quality is

The victims: Code quality

- What is code quality and why is it important
 - Code quality is
 - Code do what it should do (functioning)

The victims: Code quality

- What is code quality and why is it important
 - Code quality is
 - Code do what it should do (functioning)
 - Easy to check does it do what it should do (testable)

The victims: Code quality

- What is code quality and why is it important
 - Code quality is
 - Code do what it should do (functioning)
 - Easy to check does it do what it should do (testable)
 - Readable (might be the most important)

The victims: Code quality

- What is code quality and why is it important
 - Code quality is
 - Code do what it should do (functioning)
 - Easy to check does it do what it should do (testable)
 - Readable (might be the most important)
 - This is important because

The victims: Code quality

- What is code quality and why is it important
 - Code quality is
 - Code do what it should do (functioning)
 - Easy to check does it do what it should do (testable)
 - Readable (might be the most important)
 - This is important because
 - It is a base for team work

The victims: Code quality

- What is code quality and why is it important
 - Code quality is
 - Code do what it should do (functioning)
 - Easy to check does it do what it should do (testable)
 - Readable (might be the most important)
 - This is important because
 - It is a base for team work
 - It is a base for later product support and extension

The victims: Code quality

- Why the code quality is a victim

The victims: Code quality

- Why the code quality is a victim
 - No time for refactoring

The victims: Code quality

- Why the code quality is a victim
 - No time for refactoring
 - Refactoring is an essential part of the development

The victims: Code quality

- Why the code quality is a victim
 - No time for refactoring
 - Refactoring is an essential part of the development
 - Especially if the task is not clear

The victims: Code quality

- Why the code quality is a victim
 - No time for refactoring
 - Refactoring is an essential part of the development
 - Especially if the task is not clear
 - Planning was sacrificed, you know

The victims: Code quality

- Direct damage

The victims: Code quality

- Direct damage
 - Code is expensive to maintain and extend

The victims: Code quality

- Direct damage
 - Code is expensive to maintain and extend
 - Will be important if and only if the project will survive

The victims: Code quality

- Direct damage
 - Code is expensive to maintain and extend
 - Will be important if and only if the project will survive
 - Code readability

The victims: Code quality

- Direct damage
 - Code is expensive to maintain and extend
 - Will be important if and only if the project will survive
 - Code readability
 - Testing

The victims: Code quality

- Direct damage
 - Code is expensive to maintain and extend
 - Will be important if and only if the project will survive
 - Code readability
 - Testing
- Collateral damage

The victims: Code quality

- Direct damage
 - Code is expensive to maintain and extend
 - Will be important if and only if the project will survive
 - Code readability
 - Testing
- Collateral damage
 - The team spirit

The victims: Code quality

- Direct damage
 - Code is expensive to maintain and extend
 - Will be important if and only if the project will survive
 - Code readability
 - Testing
- Collateral damage
 - The team spirit
 - Difficult to extend the team

The victims: Architecture

- Why

The victims: Architecture

- Why
 - No direct profit

The victims: Architecture

- Why
 - No direct profit
 - Slow down the things

The victims: Architecture

- Why
 - No direct profit
 - Slow down the things
 - Wrong

The victims: Architecture

- Direct damage

The victims: Architecture

- Direct damage
 - No pivot: almost all the business decisions are final and not reversible

The victims: Architecture

- Direct damage
 - No pivot: almost all the business decisions are final and not reversible
- Collateral damage

The victims: Architecture

- Direct damage
 - No pivot: almost all the business decisions are final and not reversible
- Collateral damage
 - No examination for the business decisions and timeline

The victims: Development cycle

- The usual development cycle: regular reviews, regular refactoring, bugfixing, etc

The victims: Development cycle

- The usual development cycle: regular reviews, regular refactoring, bugfixing, etc
- Why

The victims: Development cycle

- The usual development cycle: regular reviews, regular refactoring, bugfixing, etc
- Why
 - It is really unpleasant to read a dirty code

The victims: Development cycle

- The usual development cycle: regular reviews, regular refactoring, bugfixing, etc
- Why
 - It is really unpleasant to read a dirty code
 - It is really unpleasant to get the WTF comments on your code

The victims: Development cycle

- Direct damage

The victims: Development cycle

- Direct damage
 - Bus factor

The victims: Development cycle

- Direct damage
 - Bus factor
- Collateral damage

The victims: Development cycle

- Direct damage
 - Bus factor
- Collateral damage
 - Team spirit

The victims: Development cycle

- Direct damage
 - Bus factor
- Collateral damage
 - Team spirit
 - This one can become technical one day

The victims: Learning cycle

- Why

The victims: Learning cycle

- Why
 - Lack of time

The victims: Learning cycle

- Why
 - Lack of time
 - And all of the things you need to learn effectively

The victims: Learning cycle

- Why
 - Lack of time
 - And all of the things you need to learn effectively
 - No direct profit

The victims: Learning cycle

- Direct damage

The victims: Learning cycle

- Direct damage
 - Stagnation in the project

The victims: Learning cycle

- Direct damage
 - Stagnation in the project
 - No staff qualification grow

The victims: Learning cycle

- Direct damage
 - Stagnation in the project
 - No staff qualification grow
- Collateral damage

The victims: Learning cycle

- Direct damage
 - Stagnation in the project
 - No staff qualification grow
- Collateral damage
 - Nothing to bring with you after the project ends

The victims: Learning cycle

- Direct damage
 - Stagnation in the project
 - No staff qualification grow
- Collateral damage
 - Nothing to bring with you after the project ends
 - Work is boring

The victims: Learning cycle

- Direct damage
 - Stagnation in the project
 - No staff qualification grow
- Collateral damage
 - Nothing to bring with you after the project ends
 - Work is boring
 - This one can become technical

The victims: System of values

- System of values deformation

The victims: System of values

- System of values deformation
 - Not really technical but extremely important

The victims: System of values

- System of values deformation
 - Not really technical but extremely important
- Why

The victims: System of values

- System of values deformation
 - Not really technical but extremely important
- Why
 - The people who write fast are “better” than the people who write readable

The victims: System of values

- System of values deformation
 - Not really technical but extremely important
- Why
 - The people who write fast are “better” than the people who write readable
 - The whole project view is not welcomed

The victims: System of values

- System of values deformation
 - Not really technical but extremely important
- Why
 - The people who write fast are “better” than the people who write readable
 - The whole project view is not welcomed
 - Mind your business approach

The victims: System of values

- Direct damage

The victims: System of values

- Direct damage
 - Morale of your units decreases

The victims: System of values

- Direct damage
 - Morale of your units decreases
 - The other problems mentioned above become permanent

The victims: System of values

- Direct damage
 - Morale of your units decreases
 - The other problems mentioned above become permanent
 - No even one step ahead view

The victims: System of values

- Direct damage
 - Morale of your units decreases
 - The other problems mentioned above become permanent
 - No even one step ahead view
 - The task considered completed as early as possible

The victims: System of values

- Direct damage
 - Morale of your units decreases
 - The other problems mentioned above become permanent
 - No even one step ahead view
 - The task considered completed as early as possible
 - Really important but not urgent stuff continuously delayed forever

The victims: System of values

- Direct damage
 - Morale of your units decreases
 - The other problems mentioned above become permanent
 - No even one step ahead view
 - The task considered completed as early as possible
 - Really important but not urgent stuff continuously delayed forever
- Collateral damage

The victims: System of values

- Direct damage
 - Morale of your units decreases
 - The other problems mentioned above become permanent
 - No even one step ahead view
 - The task considered completed as early as possible
 - Really important but not urgent stuff continuously delayed forever
- Collateral damage
 - There will be a negative selection in your team

The victims: Technical debt

- I do not see it as a part of the problem

The victims: Technical debt

- I do not see it as a part of the problem
 - Consider it as a cheap loan which will be never sold to collectors

The victims: Technical debt

- I do not see it as a part of the problem
 - Consider it as a cheap loan which will be never sold to collectors
- As with the other types of loans the key thing is to keep it under control

The victims: Technical debt

- I do not see it as a part of the problem
 - Consider it as a cheap loan which will be never sold to collectors
- As with the other types of loans the key thing is to keep it under control
- But the things mentioned above can induce the uncontrolled debt grow

The victims: Technical debt

- I do not see it as a part of the problem
 - Consider it as a cheap loan which will be never sold to collectors
- As with the other types of loans the key thing is to keep it under control
- But the things mentioned above can induce the uncontrolled debt grow
- Also it might become hard to pay off

What we can do: Planing

- Insist

What we can do: Planing

- Insist
 - At least definition of done must be provided

What we can do: Planing

- Insist
 - At least definition of done must be provided
 - “I want to see this and this” is way better than “do this and this”

What we can do: Planing

- Insist
 - At least definition of done must be provided
 - “I want to see this and this” is way better than “do this and this”
- Do your own planing

What we can do: Planing

- Insist
 - At least definition of done must be provided
 - “I want to see this and this” is way better than “do this and this”
- Do your own planing
 - Mostly to be aware of the “plans vs reality” conflict

What we can do: Planing

- Request task to be created in your task tracker before task is started

What we can do: Planing

- Request task to be created in your task tracker before task is started
 - Review the task and request DoD to be clear

What we can do: Planing

- Request task to be created in your task tracker before task is started
 - Review the task and request DoD to be clear
 - Request task description to be updated in case task itself got updated

What we can do: Code quality

- Make the “code quality” term formal:
provide the document to the team

What we can do: Code quality

- Make the “code quality” term formal: provide the document to the team
- Do the review yourself

What we can do: Code quality

- Make the “code quality” term formal:
provide the document to the team
- Do the review yourself
 - Provide as detailed comments as possible

What we can do: Code quality

- Make the “code quality” term formal: provide the document to the team
- Do the review yourself
 - Provide as detailed comments as possible
- Do “look at this good one” sessions based on the project code

What we can do: Code quality

- Make the “code quality” term formal: provide the document to the team
- Do the review yourself
 - Provide as detailed comments as possible
- Do “look at this good one” sessions based on the project code
 - Just a broadcast message with a link to PR

What we can do: Code quality

- Make the “code quality” term formal: provide the document to the team
- Do the review yourself
 - Provide as detailed comments as possible
- Do “look at this good one” sessions based on the project code
 - Just a broadcast message with a link to PR
- Request unit tests

What we can do: Code quality

- Make the “code quality” term formal: provide the document to the team
- Do the review yourself
 - Provide as detailed comments as possible
- Do “look at this good one” sessions based on the project code
 - Just a broadcast message with a link to PR
- Request unit tests
 - Absolute coverage value is not so good metric

What we can do: Code quality

- Make the “code quality” term formal: provide the document to the team
- Do the review yourself
 - Provide as detailed comments as possible
- Do “look at this good one” sessions based on the project code
 - Just a broadcast message with a link to PR
- Request unit tests
 - Absolute coverage value is not so good metric
 - Change of the coverage is better: shouldn't decrease

What we can do: Architecture

- Insist

What we can do: Architecture

- Insist
 - Insist again

What we can do: Architecture

- Insist
 - Insist again
 - It is ok to have an architect as a contractor

What we can do: Architecture

- Insist
 - Insist again
 - It is ok to have an architect as a contractor
 - Request C4 model diagrams

What we can do: Architecture

- Insist
 - Insist again
 - It is ok to have an architect as a contractor
 - Request C4 model diagrams
- Do the arch yourself if you are capable

What we can do: Architecture

- Insist
 - Insist again
 - It is ok to have an architect as a contractor
 - Request C4 model diagrams
- Do the arch yourself if you are capable
 - Use C4 model

What we can do:

Development cycle

- This is a routine could save you and your project, so keep it by any means

What we can do:

Development cycle

- This is a routine could save you and your project, so keep it by any means
- Make the reviewer responsible for the problems caused by deploy (if any) as well as author

What we can do:

Development cycle

- This is a routine could save you and your project, so keep it by any means
- Make the reviewer responsible for the problems caused by deploy (if any) as well as author
 - Ideally reviewer should be able to fix the problem as effective as an author

What we can do:

Development cycle

- This is a routine could save you and your project, so keep it by any means
- Make the reviewer responsible for the problems caused by deploy (if any) as well as author
 - Ideally reviewer should be able to fix the problem as effective as an author
- Ask the teammates to be extremely polite in their comments

What we can do:

Development cycle

- This is a routine could save you and your project, so keep it by any means
- Make the reviewer responsible for the problems caused by deploy (if any) as well as author
 - Ideally reviewer should be able to fix the problem as effective as an author
- Ask the teammates to be extremely polite in their comments
 - Make them polite

What we can do:

Development cycle

- This is a routine could save you and your project, so keep it by any means
- Make the reviewer responsible for the problems caused by deploy (if any) as well as author
 - Ideally reviewer should be able to fix the problem as effective as an author
- Ask the teammates to be extremely polite in their comments
 - Make them polite
 - Do not ask me how

What we can do: Learning cycle

- Really hard to fix

What we can do: Learning cycle

- Really hard to fix
- Add the “what new did I learn this sprint” to the retrospective if you can

What we can do: Learning cycle

- Really hard to fix
- Add the “what new did I learn this sprint” to the retrospective if you can
 - Do not make it formal or frustrating!

What we can do:

System of values

- Out of your control

What we can do: System of values

- Out of your control
- Hire those who can resist this leprosy

What we can do: Tech debt

- Write down every single piece at the time you realise you are adding it

Conclusions

- TTM dependent projects are tough

Conclusions

- TTM dependent projects are tough
 - They are hard to manage and hard to keep the damage under control

Conclusions

- TTM dependent projects are tough
 - They are hard to manage and hard to keep the damage under control
 - Forewarned is forearmed

Conclusions

- TTM dependent projects are tough
 - They are hard to manage and hard to keep the damage under control
 - Forewarned is forearmed
- Keep in mind the way you can be damaged and the way you can mitigate it

Conclusions

- TTM dependent projects are tough
 - They are hard to manage and hard to keep the damage under control
 - Forewarned is forearmed
- Keep in mind the way you can be damaged and the way you can mitigate it
- Non-TTM-dependent projects can have all the same problems

Conclusions

- TTM dependent projects are tough
 - They are hard to manage and hard to keep the damage under control
 - Forewarned is forearmed
- Keep in mind the way you can be damaged and the way you can mitigate it
- Non-TTM-dependent projects can have all the same problems
- Keep calm and carry on

Thank you!

Questions?



MICROAVIA



Co-organizer

Yandex